

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

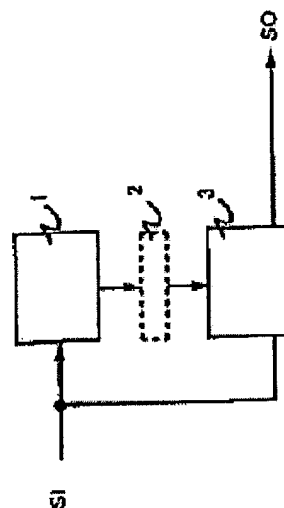
EVENT DRIVING TYPE VEHICLE CONTROL COMPUTER

Patent number: JP4118704
Publication date: 1992-04-20
Inventor: YAMAMOTO AKITO; others: 07
Applicant: NISSAN MOTOR CO LTD; others: 01
Classification:
- **international:** G05B15/02; F02D45/00
- **europaean:**
Application number: JP19900237153 19900910
Priority number(s):

Abstract of JP4118704

PURPOSE:To detect the even at a high speed, and to facilitate the development of a program and the maintenance by providing an event processing processor used exclusively for the even processing, and detecting the even generation independently from an arithmetic period in a control operation processor.

CONSTITUTION:An even processing processor 1 detects which event is generated, based on an input signal SI from various sensors. That is, the event processing processor 1 detects the generation of a transient in arbitrary data on an address, discriminates a set of transients in which the generation of the transient is aligned in a prescribed sequence, detects which of the sets of transients corresponding to plural events, respectively determined in advance it is, and outputs it as generation of the prescribed even. In such a way, the even processing corresponding to the generated even can be executed quickly, and also, the control program development is facilitated.



Data supplied from the **esp@cenet** database - Patent Abstracts of Japan

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A) 平4-118704

⑬ Int. Cl.⁵

識別記号

庁内整理番号

⑭ 公開 平成4年(1992)4月20日

G 05 B 15/02
F 02 D 45/00

M
3 7 4

7740-3H
8109-3G

審査請求 未請求 請求項の数 2 (全13頁)

⑮ 発明の名称 イベント駆動型車両制御用コンピュータ

⑯ 特 願 平2-237153

⑰ 出 願 平2(1990)9月10日

⑱ 発 明 者 山 本 明 人 神奈川県横浜市神奈川区宝町2番地 日産自動車株式会社
内

⑲ 発 明 者 安 保 敏 巳 神奈川県横浜市神奈川区宝町2番地 日産自動車株式会社
内

⑳ 発 明 者 南 英 洋 神奈川県横浜市神奈川区宝町2番地 日産自動車株式会社
内

㉑ 出 願 人 日産自動車株式会社 神奈川県横浜市神奈川区宝町2番地

㉒ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

㉓ 代 理 人 弁理士 中村 純之助 外1名

最終頁に続く

明 糸田 三

1. 発明の名称

イベント駆動型車両制御用コンピュータ

2. 特許請求の範囲

1. 車両の各種運転状態や運転者の操作状態を検出するセンサの信号を入力し、該信号に基づいて車両の制御に必要な各種制御演算を行ない、該演算結果に基づいた制御信号を車両制御用の各種アクチュエータに送出する車両制御用コンピュータにおいて、

運転者の操作や車両の運転状態の変化、すなわちイベントをアドレス上のデータから検出するイベント処理用プロセッサと、

該イベント処理用プロセッサの検出結果に基づいて演算処理を実行する制御演算用プロセッサと、を備え、

かつ、上記イベント処理用プロセッサは、アドレス上の任意のデータにおけるトランジェントの発生を検出し、上記トランジェントの発生が所定

の順序に整列したトランジェントの組を判別して、それが予め定められた複数のイベントにそれぞれ対応したトランジェントの組のうちのいずれであるかを検出し、それを当該所定のイベントの発生として出力するものであることを特徴とするイベント駆動型車両制御用コンピュータ。

2. 車両の各種運転状態や運転者の操作状態を検出するセンサの信号を入力し、該信号に基づいて車両の制御に必要な各種制御演算を行ない、該演算結果に基づいた制御信号を車両制御用の各種アクチュエータに送出する車両制御用コンピュータにおいて、

運転者の操作や車両の運転状態の変化、すなわちイベントをアドレス上のデータから検出するイベント処理用プロセッサと、

該イベント処理用プロセッサの検出結果に基づいて演算処理を実行する制御演算用プロセッサと、を備え、

かつ、上記イベント処理用プロセッサは、アドレス上の任意のデータにおけるトランジェントの

発生を検出し、上記トランジェントの発生が所定の順序に整列したトランジェントの組を判別して、それが予め定められた複数のイベントにそれぞれ対応したトランジェントの組のうちのいずれであるかを検出し、それを当該所定のイベントの発生として出力するものであり、

さらに、上記イベント処理用プロセッサの判別結果、すなわちイベントの発生とその種類とを受けて上記制御演算用プロセッサの複数のプログラムの実行管理を行なう制御手順管理手段を備えたことを特徴とするイベント駆動型車両制御用コンピュータ。

3. 発明の詳細な説明

〔発明の利用分野〕

本発明は、車両の制御機器に用いるコンピュータに関し、特に、運転者の操作や車両の運転状態の変化、すなわちイベントを速やかに検出し、これに即座に反応して演算処理を行ない、最適な制御操作を時間遅れなく実行するコンピュータに関する。

なわちイベント処理を、定周期割込みルーチンで実行するか、もしくはバックグラウンドでのプログラムの実行ループで行なわざるを得なかった。

そのため実際のイベントの発生からイベントの検出およびその処理までに時間的な遅れが生じ、車両の制御特性、例えばエンジンの排気浄化性能や燃費性能の悪化を招き、また運転者にとって不快な応答遅れや予期せぬ反応を引き起こす原因になっている。さらに、微妙なタイミングのずれによって制御機器の動作が大きく異なるため、外部からの制御機器の検証を困難にし、そのうえ実現のためにいたずらに煩雑なプログラムロジックを必要とするため、制御プログラムを開発する際に多大の工数を必要とし、またこのような時間的空間的に分断された論理の流れは人間の思考特性に一致しないため、プログラム作成の際にミスを生じ易く、信頼性の低下を招くおそれがある。また、そのようなプログラムは第三者による判読を困難にしてメンテナンス性の悪化を招く等、多大の悪影響を与えている。

〔従来技術〕

マイクロコンピュータを用いて車両の各種制御を行なう装置に関しては、従来、種々のものが提案されている（例えば、特開昭55-60639号、特開昭55-134732号、特開昭54-58116号等、多数あり）。

車両用の制御装置においては、運転者の操作や車両の運転状態の変化、すなわちイベントを即座に検出し、これに応じた制御を即座に実行する必要がある。なお、上記のごとき意味で“イベント”という語を使用した例としては、「アイイーイーイー トランザクションズ オン ソフトウェア エンジニアリング (IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-6, NO. 1, JANUARY 1980, "Specifying Software Requirements for Complex System : New Techniques and Their Application")」がある。

しかし、従来のマイクロコンピュータを用いた制御装置においては、本来、上記のようなイベントに即座して実行されなければいけない処理、す

以下、上記のごとき従来技術の問題点を事例に即して詳細に説明する。

第8図は車両のエンジン制御における燃料噴射制御の一例のブロック図である。

第8図に示すように、一般的なエンジン制御系においては、基本噴射量演算部51によってエンジン回転数 N と吸入空気量 Q とに基づいて基本噴射量 Tp を演算する。そして定常的には、この基本噴射量 Tp に対応した燃料をエンジンの回転基準信号（エンジンの1回転または2回転毎に所定クランク角度で発生するパルス信号）に同期して各気筒に噴射することにより、エンジンに対する燃料供給を制御している。

また、エンジンの加減速時には、補正量演算部52において、スロットルポジションセンサの信号またはスロットル全開スイッチ（いわゆるアイドルスイッチ）の信号 SS からエンジンの加減速を検出し、回転数 N 、吸入空気量 Q および冷却水温 Tw に対応した補正噴射量 Ti を算出し、噴射量演算部53で前記基本噴射量 Tp と補正噴射量 Ti

を加算または減算して燃料噴射量 T_e を算出し、それに応じた燃料噴射を行なうことにより、燃料噴射量の増減補正を行なって過渡時の燃料供給量の最適化を図っている。

第9図は、上記の補正量演算部52における過渡時の補正機能を示すブロック図である。

第9図において、まず、変化検出部54でアイドルスイッチ信号SSのオン/オフの変化を検出する。この例ではアイドルスイッチがオンからオフになったとき、すなわちスロットルが全閉状態から開の状態に移った時点判定している。これは第9図の演算が一定周期(例えば10ms毎)で実行されるのを利用して、前回の信号SSの読み値と今回の読み値を比較し、前回がオンで今回がオフという条件を検出しようとするものである。

次に、割込み噴射量演算部55では、前回オンで今回オフの条件が検出された場合、すなわちスロットル弁が全閉状態から開状態に変化した場合にのみ割込み噴射処理を実行する。割込み噴射処理ではエンジンの回転数 N 、吸入空気量 Q および

冷却水温 T_w に応じて所定の演算を行ない、割込み噴射すべき燃料の噴射量を算出する。

次に、噴射判定部56では、噴射すべき燃料の時間換算値と演算終了時点でのクランク角度から燃料噴射を開始すべきタイミングを算出し、これと吸気バルブの開閉タイミングと通常噴射のオン/オフのタイミングとを比較して即時噴射すべきか否かを判定する。すなわち、今回の吸気バルブ開のタイミング中に噴射が終了できる場合には即時噴射し、噴射が間にあわない場合には次の吸気バルブ開のタイミングまで噴射を保留する。57は即時噴射の処理、58は噴射保留の処理である。

次に、上記の噴射タイミングについて説明する。

本来、燃料噴射は、吸気弁が開いた時点から次に吸気弁が閉じる時点までの間に特定の気筒の特定の燃焼サイクルに必要な燃料の噴射を完了していなければならない。

ところで、通常インジェクタから噴射された燃料は、数センチ(5~10センチ)程度空中を飛

行した後、吸気弁に到達するが、この間の燃料の飛行速度が例えば10m/s程度であるため、上記の距離を飛行する間に5~10msの時間遅れが生じる。またインジェクタの燃料噴射量の制御は時間パルスに換算されて行なわれるため、一定の燃料量の噴射を完了するためには噴射時間そのものによる遅れが加わる。

従って、特定の気筒の特定の燃焼サイクルに間に合うように燃料を緊急に噴射するためには、噴射に必要な噴射パルス時間を T_{int} 、飛行に必要な時間を T_{flt} 、噴射弁の応答遅れを T_s とすれば、最低 $T_{int}+T_{flt}+T_s$ 時間だけ吸気弁の閉じるタイミングに先立って噴射を開始しなければならない。例えば T_{int} を5ms、 T_{flt} を7.5ms程度(平均的にこの程度に設定される)、 T_s を1msとすれば、燃料噴射の機制的制約のみを考慮しても、最低13.5msを要することになる。

これに対して、アイドル状態からスロットル開状態に変化した時における空気のシリンダへの吸入遅れは、高性能エンジンでは20~30ms程度

なので、前記の燃料噴射の遅れ時間を考慮すると、本質的にタイミング上きわめて厳しい制約を満たさなくてはならないことが判る。

ところが従来のエンジンの制御系にあっては、アイドルスイッチのオンオフの判定のプログラム(例えば後記第10図)の実行タイミングは、一定の時間間隔(以下、演算周期と記す。例えば10ms程度の値)毎であるため、アイドルスイッチの変化のタイミングと演算周期の関係によっては、オン/オフ判定は、次の演算周期すなわち最大10ms後までずれ込むことになる。

従って前記のごとき機制的制約13.5msに加えて、さらに10msの遅れが加わった場合には、合計23.5msを要することになり、しかも通常これにコンピュータの演算時間による遅れが加わることになるので、特定の燃焼サイクルに間に合うように燃料供給を行なうのはほとんど不可能となってしまう。

上記の遅れ時間10msは、エンジンの回転角度に換算すると、6000rpmでは360度(1回

転分)、最も低回転であるアイドリング時の600rpmでも36度(1回転の1/10)に相当する。したがって、例えば、前記のごときスロットルスイッチ信号の変化の検出が遅れて吸気弁の開のタイミングを逃し、それによって燃料供給が次の燃焼サイクルにずれ込むという事態が生じ、そのため、燃焼に必要な空燃比が確保できなくなり、失火や不整燃焼を起こしてエンジンの排気浄化、燃費、運転性能に重大な悪影響を与えるおそれがある。

また、他の条件を全く同一と仮定しても、アイドルスイッチの変化のタイミングがわずかに数マイクロ秒(アイドルスイッチを読み込むインストラクションの実行時間)異なっただけでも、オン/オフ判定が次の10ms後の演算周期までずれ込み、制御プログラムの実行タイミング、従って制御出力そのものが大幅に異なるということは、制御プログラムの外部からの検証にも重大な悪影響を与えることは容易に理解されよう。

なお、ここで述べた例のように、入力信号が単

純なオン/オフ信号の場合には、この信号を直接に割込み信号とし、いわゆる割込み処理として演算を実行することも原理的には可能である。

しかし、車両の制御系でイベント検出に用いられるデータは、アナログ信号、およびそれをA/D変換してデジタル化したデータ列、あるいは入力データ列に所定の演算を加えた結果のデータ列など様々であり、それらの全てについて時間的な制約は共通であるため、イベント検出のためには、入力信号に限らずメモリ上の多数のデータ(1bit〜数ワードの単位)のトランジェントを検出する必要があり、しかもその総数はエンジン制御を例にとると100を越えるため、これらの検出をすべて割込み処理を用いて行なうのは、下記の理由によって実際には不可能である。すなわち、割込み数が多くなると、割込みの発生要因を判別するプログラムの実行時間とレジスタの追跡処理の時間が多くなり、制御プログラムの実行効率がきわめて悪く、より高速の応答を要求されるパルス信号やアナログ信号の計測処理に誤差や検

出遅れによる誤動作を引き起こす結果となってしまうためである。

第10図は、前記第9図の機能をプログラムで実行した場合のフローチャートである。

第10図において、(a)から(e)までが第9図の5.4に、(f)が5.5に、(g)が5.6に、(h)が5.7に、(i)が5.8に相当する。

第10図で行なっているのは、最も単純な1ビットの信号のオン・オフの判定であるが、このためだけに(a)から(e)までのロジックを必要としている。しかも(c)、(d)、(e)から判るように、同一の信号の同一の状態が前の演算周期における信号の状態によって異なった意味を持ったため、煩雑で非常に認識・理解のしにくい流れとなっている。

このプログラムの煩雑さは、一見、当然のこととして見逃されがちであるが、必然的にプログラム行数を増加させるだけでなく、開発にあたる人間の認知科学的・心理的特性と相容れない性質を持つため、ミスの発見を困難とし、開発期間、開

発コストに悪影響を及ぼし、ひいては製品の信頼性の低下を招きかねない等、非常に大きな意味を持っているといわなければならない。

〔発明が解決しようとする課題〕

以上説明したごとく、従来のコンピュータを用いた車両用制御装置においては、

①実際のイベントの発生からイベントの検出およびその処理までに時間的な遅れが生じ、そのため車両の制御特性の悪化を招き、また運転者にとって不快な応答遅れや予期せぬ反応を引き起こす原因となる。

②微妙なタイミングのずれによって制御機器の動作が大きく変わるため、外部からの制御機器の検証を困難にしている。

③いたずらに煩雑なプログラムロジックを必要とするため、制御プログラムを開発する際に多大の工数を必要とする。

④プログラム論理の流れが人間の思考特性に一致しないため、プログラム作成の際にミスを生じやすく、信頼性の低下を招く恐れがある。

⑤プログラムの第三者による判読が困難になるので、メンテナンス性の悪化を招く、等の種々の問題があった。

本発明は、上記のごとき従来技術の問題を解決するためにさなれたものであり、イベント発生時に、発生したイベントに対応したイベント処理を迅速に実行することが出来、しかも、いたずらに煩雑なプログラムロジックを必要とすることなく、制御プログラム開発の容易なイベント駆動型車両制御用コンピュータを提供することを目的とする。
〔課題を解決するための手段〕

上記の目的を達成するため、本発明においては、特許請求の範囲に記載するように構成している。

第1図は本発明の機能ブロック図である。

第1図において、1はイベント処理用プロセッサであり、各種センサからの入力信号SIに基づいて、どのイベントが発生したかを検出する。

すなわち、上記イベント処理用プロセッサは、アドレス上の任意のデータにおけるトランジェントの発生を検出し、上記トランジェントの発生が

行する。

なお、上記制御手順管理手段3は、専用のハードウェアとして構成することも出来るが、イベント処理用プロセッサまたは制御演算用プロセッサ内のプログラム上の機能として構成することも出来る。

〔作用〕

上記のように、本発明においては、演算処理を実行する制御演算用プロセッサの他に、イベント処理専用のイベント処理用プロセッサを備え、イベントの発生を制御演算用プロセッサにおける演算周期とは独立に検出するように構成している。

すなわち、イベント処理用プロセッサは、アドレス上の任意のデータにトランジェント（遷移）が発生した場合に、該トランジェントの発生が所定の順序に整列したトランジェントの組を検出し、そのトランジェントの組が予め定められたどのイベントに対応したものであるかを判別して、それを当該所定のイベントの発生として出力するものである。

所定の順序に整列したトランジェントの組を判別して、それが予め定められた複数のイベントにそれぞれ対応したトランジェントの組のうちのいずれであるかを検出し、それを当該所定のイベントの発生として出力するものである。

また、2は制御演算用プロセッサであり、通常は所定のプログラムに従って制御演算を行ない、その結果として得られた制御用の出力信号SOを出力する。また、イベント発生時にはイベント処理用プロセッサ1の検出結果に基づいて、その発生したイベントに対応した制御プログラムの演算を実行し、その結果として得られたイベント発生時の出力信号を出力する。

また、破線で示した制御手順管理手段3は、第2請求項に対応するものであり、イベント処理用プロセッサ1の判別結果、すなわちイベントの発生とその種類とを受けて制御演算用プロセッサ2の複数のプログラムの実行管理を行なう。この場合、制御演算用プロセッサ2は、上記制御手順管理手段3で指定された制御プログラムの演算を実

上記のように構成したことにより、従来のように制御演算用プロセッサで演算周期に従ってイベントの検出を行なう必要がなくなるので、イベントの検出を高速で行なうことが出来、かつ、膨大な割込みの発生によって他の演算に支障を来すおそれもない。また、イベント検出用のプログラムも前記第10図の例のような煩雑で認識・理解のしにくいものは unnecessary になるので、プログラムの開発やメンテナンスも容易になる。

また、イベント処理用プロセッサの判別結果、すなわちイベントの発生とその種類とを受けて制御演算用プロセッサの複数のプログラムの実行管理を行なう制御手順管理手段を備えることにより、イベントの種類に対応した制御演算を確実に実行させることが出来る。

〔発明の実施例〕

第2図は、本発明の一実施例のブロック図である。

第2図において、11はイベント処理用プロセッサ、12は制御演算用プロセッサである。また、

RAM13およびROM14はそれぞれイベント処理用プロセッサ11専用のメモリ、RAM15およびROM16はそれぞれ制御演算用プロセッサ12専用のメモリである。また、RAM17はイベント処理用プロセッサ11と制御演算用プロセッサ12との共用のメモリであり、両プロセッサから同じ情報を読み書きできるものである。

また、SI₁～SI₁は各種センサからの入力信号であり、例えばエンジンの吸入空気量信号、回転速度信号、冷却水温信号、スロットル開度信号、アイドルスイッチ信号、空燃比信号等である。また、SO₁～SO₁は各種アクチュエータへの出力信号であり、例えば燃料噴射信号、ブレーキ制御信号、電子制御変速機への変速信号等である。

また、入出力装置18は、上記の各種入力信号SI₁～SI₁を入力（必要がある場合はA/D変換後に入力）し、かつ、演算結果の各種出力信号SO₁～SO₁を出力する。

また、上記の各構成要素間の信号の授受は、メインバス19を介して行なわれる。

イベント発生信号として出力する。

また、7はイベント判定手段であり、ランジェントの発生順序が予め定められた何のイベントに対応するものであるかを判別して、最終的にイベント検出信号として出力するものである。

なお、このイベント処理用プロセッサは、内部的に直列に接続され、さらに複雑なイベント発生シーケンスを検出する構成とすることも可能である。

また、8はイベント管理手段であり、上記5～7の連携動作をイベント管理プログラム9によって制御する。

また、イベント管理プログラム9は、例えば第3図(B)に示すときのものである。この例は、アイドルスイッチがオンからオフになった場合に“加速イベント”であることを判別する場合を例示している。

次に、上記の加速イベントを例として作用を説明する。

まず、運転者のアクセル操作は、スロットル全

次に、第3図(A)は、上記のイベント処理用プロセッサ11における演算内容の一実施例をブロックで示した図であり、第3図(B)はイベント管理プログラム9の一例図である。

以下、第3図に基づいてイベント処理用プロセッサの動作を説明する。

第3図において、5はデジタル比較手段であり、メモリ(RAM17)上の任意のビット数の所定のデータ(ビット、ワード等の単位)を比較する。このデジタル比較手段5は、所定のアドレスのメモリに対する書き込みが行なわれたことを検出するアドレス一致検出手段5-1と、イベント管理プログラム9中の所定値とメモリ上のデータとを比較し、その結果を状態コードとして出力する比較手段5-2から構成される。

また、6はランジェント検出手段であり、デジタル比較手段5の出力、メモリ上の所定のビット、もしくは入力信号の状態を入力して過去の状態と比較し、データの状態変化の方向(オンオフ、オフオン、両方向等)を検出し、ランジ

閉を示すスイッチ信号(アイドルスイッチ)のオン/オフとしてメモリに書き込まれる。なお、第3図では図示していないが、前記第2図の入出力装置18が入力レジスタのデータを入力レジスタが割り付けられたメモリへ転送する。この結果、メモリに対する書き込み操作が行なわれ、アイドルスイッチのアドレスにWR信号が発生する。

この信号を受けて、アドレス一致検出手段5-1はイベント管理プログラム9で指定されたアドレスと一致するものがあるか否かを検出する。そして一致が検出された場合は、通常はイベント管理プログラム中の比較データとメモリ上の一致が検出されたアドレスのデータとを比較手段5-2で比較するが、この例の場合はビットデータであるため、比較手段5-2は用いられず、直ちにランジェント検出手段6に入力される。

比較データが通常のデータであれば、アドレスの一致を検出した組の比較値とメモリ上の書き込みが行なわれたアドレスのデータとを比較手段5-2によって比較し、この比較結果が2値の出力

としてトランジェント検出手段6に入力される。

トランジェント検出手段6は、予め指定されたオン・オフの変化（全閉状態からアクセルが踏まれたトランジェント、すなわちこの場合は即イベント）を検出し、そのイベントに対応した所定の出力、すなわち、この場合は加速イベント検出用信号（accel-event）を発生する。同時に順序比較のため、この時のアイドルスイッチの状態がイベント処理用プロセッサの持つメモリ（定義されたイベント毎に確保される）に保存される。

なお、この例の場合は、単一のトランジェントがそのままイベントとして定義されているので、イベント判定手段7の動作はない。

上記のようにして、イベントの発生が検出された場合には、第1図の制御手順管理手段3内の実行予約手段（図示せず）が、この加速イベントに対して予め定められている割込み噴射量演算処理（以下、制御プロセスAと記す）の起動要求をセットする。ついで制御手順管理手段3内の実行優先

順位管理手段（図示せず）が制御プロセスの優先順位を比較し、現在実行中の処理（以後、制御プロセスBと記す）よりも制御プロセスAの実行優先順位が高ければ、制御演算用プロセッサ2に割込みを発生させ、制御プロセスBを中断して制御プロセスAの実行を開始する。制御プロセスBの優先順位がAと等しいか大きい場合は何ら操作は行なわれないが、起動の要求は保持されたままであり、制御プロセスBの実行が終了した後、確実に制御プロセスAの実行が行なわれる。なお、ここで行なわれる実行優先順位管理は、一般的なオペレーティングシステムで行なわれるものと同じであるので、詳細な説明は省略する。通常、車両の制御系では、イベント同期実行、定周期実行、その他（バックグラウンド等）の順で優先順位が高く、イベント同期実行プロセス間では先着順が原則となるため、いたずらに制御演算用プロセッサ2の実行を中断することなしに、制御プログラムの実行が可能となる。

この例の場合は、制御プロセスAでは、第4図

に示すフローチャートで割込み噴射処理を実行する。すなわち、割込み噴射処理では、エンジンの回転数N、吸入空気量Q、冷却水温Twから所定の演算を行ない、割込み噴射すべき燃料の噴射量を算出する。その後の処理は従来と同様である。

この例で判るように、イベントの検出をイベント処理用プロセッサを用いて独立して事前に行なうことにより、制御プログラムの論理の流れは、前記第10図に比較して非常に単純で理解しやすいものになっている。

なお、上記の例では、簡単のためオン・オフの入力信号の場合におけるイベント検出について説明したが、一般に運転者の操作もしくは車両の運転状態（例えばエンジンの状態）の変化、すなわちイベントは、アナログ信号もしくはデジタルのデータ列の変化として検出する必要があるが、これは前記のようにメモリ上のデータをデジタル比較手段5で所定値と比較することにより、容易に達成される。

例えば、運転者のアクセル操作がオン・オフ信号

ではなく、スロットルセンサによるアナログ信号として与えられる場合は、スロットルセンサの電圧値が変化し、これをA/D変換したデータ値がメモリに転送される。この結果、メモリ上にデータ値が書き込まれることになるが、このメモリ上のデータがデジタル比較手段5によって所定値と比較され、内部的にデジタルの2値化された状態コードに変換されることにより、オン・オフ入力信号の場合と同様に扱うことができる。

また、車両の運転状態の変化の例としては、例えばエンジン水温が変化した場合には、水温センサの電圧値が変化し、これをA/D変換したメモリ上のデータ値が変化し、以下同様にこのメモリ上のデータがデジタル比較手段5によって所定値と比較され、内部的にデジタルの状態コードに変換された後、トランジェント検出手段6によって大→小、小→大、両方向の指定に沿って状態の遷移を検出し、トランジェント検出信号を発生すれば良いことも容易に理解されよう。

次に、本発明の第2の実施例について説明する。

ここでは簡単のため、入力信号のオン/オフ時間、例えば、スロットル全閉信号のオン時間すなわちアクセルを離している時間を計測し、そのオン時間が所定時間以上継続した後にオフになった場合に加速イベントであると判定して割込み噴射量演算処理を実行する例を示す。この例はきわめて簡単な処理であるが、車両の制御系で多用される典型的な例であり、大多数の制御アルゴリズムがこの型に属する。

まず、従来において上記のごときイベント処理を行なう場合に用いたフローチャートを第11図に示す。

第11図においては、(a)、(b)、(c)、(e)の手順を経て、スロットル全閉信号がOFFからONへ変化したことを検出した場合には、(w)でON時刻の保存を行なう。なお、この検出時点までに既に演算周期分の遅れを生じることになる。

次に、アクセルが離まれてスロットル全閉信号がONからOFFに変化したことを、(a)、(b)、(c)、(d)の手順を経て検出し、(z)でOFF

判定手段7は、前記第3図と同様である。

また、21はイベントタイマ手段であり、トランジェント発生信号もしくはイベント検出信号に連動してその発生時刻を記憶するタイマである。

また、22は時間演算手段であり、イベントタイマ手段21によって保存された時刻およびメモリ上のデータに対して算術演算、論理演算を行ない、時間間隔を計算し、それが所定値より大きい場合にイベント発生信号を送出する。

また、23はイベント管理手段であり、上記5、6、7、21、22のユニットの連携動作をイベント管理プログラム24に基づいて制御する。

また、第5図(B)はイベント管理プログラムの例であり、スロットル全閉信号のオン時間が所定時間以上継続した後にオフになった場合に加速イベントであると判定するものを示す。

次に作用を説明する。

まず、前記第3図の実施例と同様に、運転者のスロットル操作は、スロットル全開を示すスイッチ信号(アイドルスイッチ)のオン/オフがメモ

時刻の保存を行ない、続いて(y)でON時間、すなわちONになってからOFFになるまでの時間間隔の算出を行なう。次に、(x)でON時間を所定値と比較し、所定値よりも長ければ加速イベントが発生したものと判断する。すなわちスロットル全閉の状態が所定時間以上継続した後に開状態に変化した場合にのみ加速イベントが発生したものと判別し、(v)の割込み噴射量演算処理を行なう。なお、割込み噴射量演算処理の内部は前記第10図の場合と同様である。

上記の例から判るように、従来の構成では、イベントの検出に煩雑なプログラムロジックが必要であった。

次に、第5図(A)は、本発明において上記の演算を行なう場合におけるイベント処理用プロセッサ11の機能を示すブロック図であり、第5図(B)はイベント管理プログラムの一例図である。

第5図において、アドレス一致検出手段5-1と比較手段5-2から構成されるデジタル比較手段5、トランジェント検出手段6およびイベント

リを通じて入力され、デジタル比較手段5を経てトランジェント検出手段6が予め指定されたオフ・オンの変化を検出してトランジェント検出信号を発生する。次に、イベント管理プログラム24の指定にしたがってトランジェント検出手段6の出力を受け、イベントタイマ手段21がこの時刻を保存する。同時にイベント処理用プロセッサ内のメモリにイベント判定手段7によってこの時のアイドルスイッチの状態を示すメモリのビットが保存される。

イベント判定手段7はトランジェント検出手段6の出力を受け、プログラムに指定された発生順序を判定し、所定の順序規則にしたがったものをイベントとして検出し、イベント管理プログラム24の演算の項に指定されたプログラムを実行する。

すなわち、この場合のようにパルス信号のオン時間を計測するには、第6図に示すような順序規則に従い、まず信号の立上り、すなわちオフ・オンのトランジェントが検出され、この時の時刻お

よび信号レベル（この場合は“1”）が保存される。

次に、信号の立下り、すなわちオン→オフのトランジェントが検出された場合には、前回の信号レベルをシフトした後、同様にこの時の時刻と信号レベル（この場合は“0”）を保存する。

次に、上記の保存された値の順序、すなわち、この場合には“1”“0”の順序が、予め定められた順序と一致しているか否かを検出し、一致を検出したときには、プログラムの演算の項目の指定に従って、時間演算手段22が時刻の減算を実行し、時間間隔を算出する。すなわち、立上り→立ち下りの順序でトランジェントが発生したときにのみ“アクセルが放された後に踏み込まれた”という複合したトランジェントと判定され、時間間隔の算出が行なわれる。更にこのプログラムの例の場合には、この時間間隔が所定値と比較され、所定値以上であった場合にのみ“アクセルが放されてから十分時間が経過してから踏まれた”という“加速イベント”発生の最終判定結果が出力さ

れる。

これら一連の動作は、イベント処理用プロセッサで行なわれ、制御演算用プロセッサにおける他の制御演算のプロセスとは独立して並列に、かつイベントの発生に従って実行されるため、原理的に検出に伴う遅れや不確定要素の影響を受けることはない。

上記のようにして加速イベントの発生が検出された後は、第7図に示すような踏み込み噴射演算処理が行なわれる。なお、第7図の処理は、加速イベント検出の内容以外は前記第4図と同様である。

この例で判るように、イベントの検出と演算に必要なデータ（この場合は時間間隔の算出）が事前に独立して完了し、“加速イベント”としての最終判定結果が通知されるため、“加速イベント”等のイベントの定義の変更による制御演算用プロセッサ側の制御プログラム（従って制御演算プロセス）の変更は皆無である。したがってイベント管理プログラムの内容によって、検出すべきイベントの種類や内容を自由に変更することが出来、

またいたずらに制御演算プロセスの実行を中断することなく制御演算用プロセッサ側のプログラムの実行管理をきわめて自由かつ容易に行なうことが出来る。

また、上記の例では、簡単のためイベント発生の時間間隔のみを計測する例を示したが、車両の制御においては、例えば或るイベントの発生から何回転経過したか（回転というイベントが何回発生したか）を判断する場合のように、イベントの発生回数そのものを計測判断する必要がある場合がある。このような例では定時間クロックではなくにイベント検出出力を計測クロックとすることにより、前記の例と同様の手法によって実現することが出来る。

また、イベント管理プログラムの指定により、異なったトランジェント又はイベントをプログラム上で扱うことにより、更に複合したイベントを処理することもできる。

〔発明の効果〕

以上述べてきたように、本発明においては、制

御演算用プロセッサとは独立にイベント処理用プロセッサを設け、制御演算処理の動作サイクルとは独立して並列にイベントの発生を検査し、かつ、イベント発生時に遅れなく制御プログラムを実行して必要な制御演算処理と制御動作を行なうように構成したことにより、車両の制御機器に必要な運転者の操作や車両の運転状態の変化を即座に検出し、遅れなく制御操作を実施することが可能となる。そのため車両の制御特性の劣化、例えばエンジンの排気浄化、燃費の悪化や運転者にとって不快な応答遅れや予期せぬ反応を引き起こす原因を取り除けるのみならず、実現のためにいたずらに煩雑なプログラムロジックを必要とすることもなくなり、制御プログラムを開発する際に必要な多大の工数を削減することが出来る。また従来のような時間的・空間的に分断された人間の思考特性に一致しない論理を排除して連続した論理の流れにできるので、ミス発生の事前に予防することが可能となり、信頼性の向上を図ることが出来、さらに第三者による判読を容易にしてメンテナン

ス性の向上を図ることが出来る。また、微少なタイミングのずれによって制御機器の動作が大きく異なる現象も排除できるため、外部からの制御機器の検証が行ない易くなる、等多くの優れた効果が得られる。

また、イベントの検出と演算に必要なデータ（時間間隔の算出やイベント数の計測）が事前に独立して完了し、かつ、イベントとしての最終判定結果が制御演算用プロセッサに通知されるため、イベントの定義の変更による制御演算用プロセッサ側の制御プログラム（従って制御演算プロセス）の変更は皆無にすることが出来るので、イベントの種類や内容を変更する場合に、プログラム開発、仕様管理、プログラムの保守に費やす膨大な工数を大幅に削減できるという効果が得られる。

さらに、いたずらに制御演算プロセスの実行を中断することなく制御演算用プロセッサ側のプログラムの実行が可能となるため、制御演算用プロセッサのもつ本来の論理的・数値的演算能力を充分に発揮させることが出来るという効果が得られ

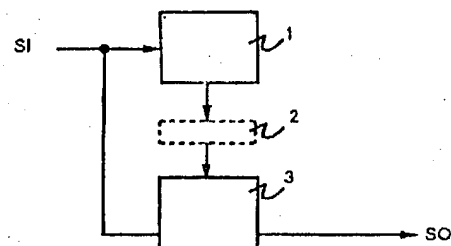
る。

4. 図面の簡単な説明

第1図は本発明の機能ブロック図、第2図は本発明の一実施例のブロック図、第3図は第2図の実施例におけるイベント処理用プロセッサの機能を示すブロック図およびイベント管理プログラムの一例図、第4図は第3図の実施例における処理手順を示すフローチャート、第5図は第2図の実施例におけるイベント処理用プロセッサの機能を示す他のブロック図およびイベント管理プログラムの他の一例図、第6図は第5図の実施例における時間計測手順を示すフローチャート、第7図は第5図の実施例における処理手順を示すフローチャート、第8図は従来のエンジン制御を示すブロック図、第9図は第8図内の一部の内容の詳細を示すブロック図、第10図は従来のイベント処理の一例を示すフローチャート、第11図は従来のイベント処理の他の例を示すフローチャートである。

<符号の説明>

- 1 … イベント処理用プロセッサ
- 2 … 制御手順管理手段
- 3 … 制御演算用プロセッサ
- 5 … デジタル比較手段
- 5-1 … アドレス一致検出手段
- 5-2 … 比較手段
- 6 … トランジェント検出手段
- 7 … イベント判定手段
- 8 … イベント管理手段
- 9 … イベント管理プログラム
- 11 … イベント処理用プロセッサ
- 12 … 制御演算用プロセッサ
- 13、15、17 … RAM 14、16 … ROM
- 18 … 入出力装置 19 … メインバス
- 21 … イベントタイマ手段 22 … 時間演算手段
- 23 … イベント管理手段
- 24 … イベント管理プログラム

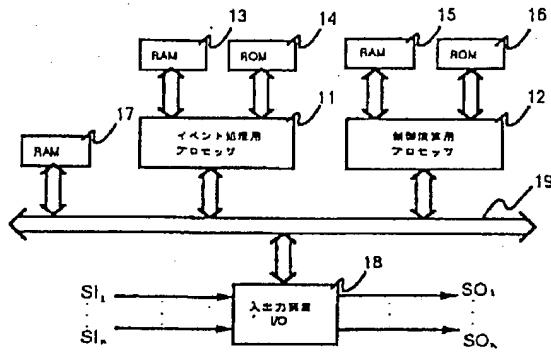


- 1: イベント処理用プロセッサ
- 2: 制御演算用プロセッサ
- 3: 制御手順管理手段
- SI: 入力信号
- SO: 出力信号

第1図

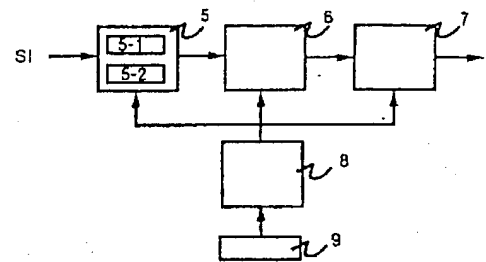
代理人弁理士 中村 純之助

代理人弁理士 和 泉 良 彦



SI₁-SI₂・・・各種センサからの入力
SO₁-SO₂・・・各種アクチュエータへの出力信号

第2図



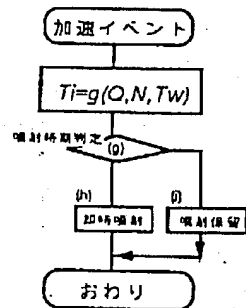
(A)

5:デジタル比較手段
5-1:アドレス一致検出手段
5-2:比較手段
6:トランジエント検出手段
7:イベント判定手段
8:イベント管理手段
9:イベント管理プログラム

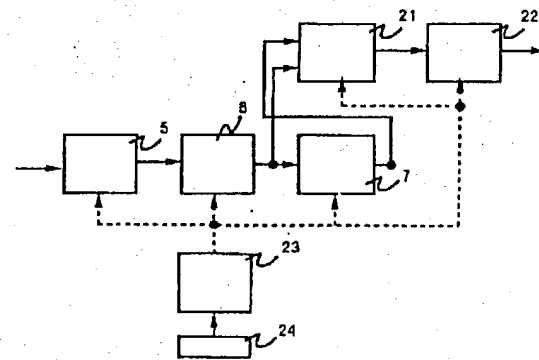
アドレス	比較値	データ型	トランジエント	検出	イベント
idle-sw		bit(0)	on-off		ACCPL_EVENT...

(B)

第3図



第4図



(A)

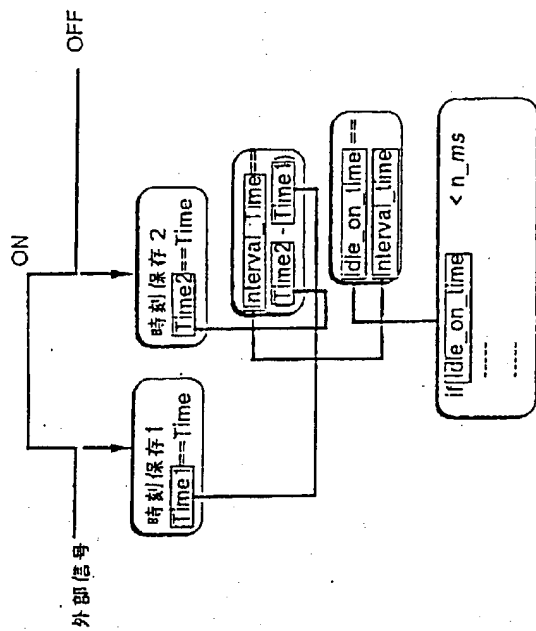
アドレス	比較値	データ型	トランジエント	検出	イベント
idle-sw		bit(0)	on-off		ACCPL_EVENT...


```

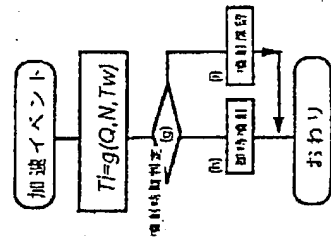
time_test
sub r2,r1
cmp r2,time_val1
bl ret
push accpl_event
ret
    
```

(B)

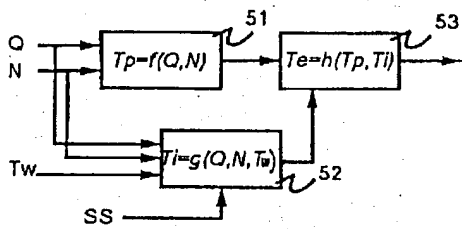
第5図



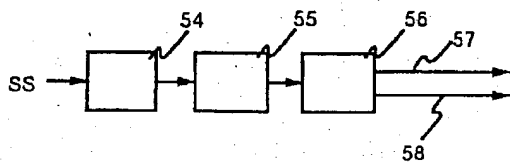
第6図



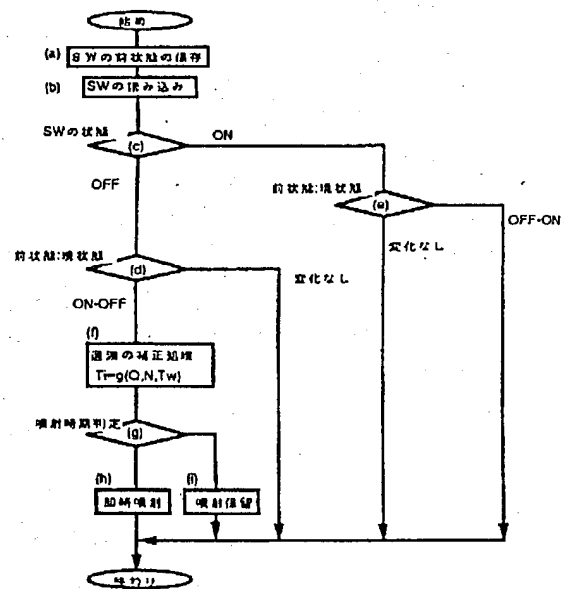
第7図



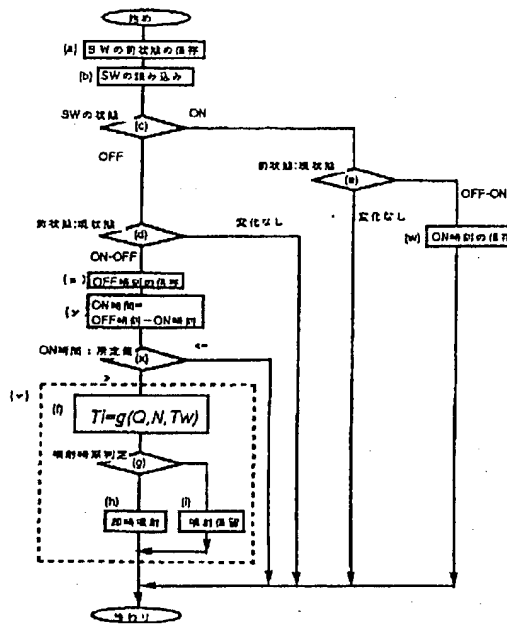
第8図



第9図



第10図



第 1 1 図

第 1 頁の続き

⑫発明者	押上	勝憲	神奈川県横浜市神奈川区宝町 2 番地 日産自動車株式会社内
⑬発明者	牧野	淳一	茨城県日立市久慈町 4026 番地 株式会社日立製作所日立研究所内
⑭発明者	栗原	伸夫	茨城県日立市久慈町 4026 番地 株式会社日立製作所日立研究所内
⑮発明者	大成	幹彦	神奈川県川崎市麻生区王禅寺 1099 番地 株式会社日立製作所システム開発研究所内
⑯発明者	志田	正実	茨城県勝田市大字高場 2520 番地 株式会社日立製作所佐和工場内